Права за достъп. Управление на достъпа до директории и файлове.

Въведение

- Система за контрол на достъпа.
- Разделени привилегии.
  - Потребител.
  - Група.
  - Всички останали.
- Режим на достъп

Контрол на достъпа до файлове и директории

- За файлове:
  - Четене (r) – преглед на съдържанието на файла;
    - Ако "другите" имат достъп, но членовете на групата нямат, то членовете на групата не могат да видят съдържанието на файла, независимо, че "другите" могат.
  - Запис (w) – промяна на съдържанието на файла;
    - Намаляване до "0".
  - Изпълнение (x):
    - За компилирани файлове (Linux binary);
    - За скриптове (изисква и права за четене).
- За директории:
  - Четене (r) – достъп до съдържанието на директорията;
    - Не можеш да разберешнищо повече за файловете;
  - Запис (w) – създаване, преименуване и изтриване на файлове;
  - Изпълнение (x) – влизане в директорията с командата "cd"
    - Има достъп до файловете, ако са известни имената им имената им;
    - Може ли да се изтрие файл, дори, ако няма право за запис на самия файл?

Проверка и промяна на правата за достъп

- За файлове: ls -l

```
$ ls -l
-rw-r--r-- 1 stoyan stoyan    42 фев  8 14:21 bginput.txt
--wxr----- 1 stoyan stoyan    51 фев  6 16:08 clock.sh
-rwxr----x 1 stoyan stoyan   184 фев  7 16:15 count.sh
-rw-r--r-- 1 stoyan stoyan    77 мар 15 16:02 cuttest
drwxr-xr-x 3 stoyan stoyan  4096 фев 26 11:46 dir1
```

- За директории: ls -ld

```
$ ls -ld my_dir*
drwxr-x--x 2 stoyan root   4096 мар 25 16:31 my_dir
drwxr-xr-x 2 stoyan stoyan 4096 фев 26 13:10 my_dir1
```

- chmod – сменя режима на достъп до файл или директория
  - chmod [<options>]<permissions><name>
  - chmod -R рекурсивно
  - chmod —reference=<name> <name> по образец
  - задаване на разрешения с цифри
    - chmod 777 file

```
u - owner    "+" включва
g - group    "-" изключва
o - other    "=" задава
a - all
        r - read
        w - write
        x - execute
```

Подробна информация за файл

```
ian@Z61t-u14:~/lpi103-2$ ls -al
total 52
drwxrwxr-x  2 ian ian 4096 Jun  8 17:09 .
drwxr-xr-x 15 ian ian 4096 Jun  8 13:26 ..
-rw-rw-r--  1 ian ian    8 Jun  8 17:02 sedtab
-rw-rw-r--  1 ian ian   24 Jun  8 13:26 text1
-rw-rw-r--  1 ian ian   25 Jun  8 13:36 text2
-rw-rw-r--  1 ian ian   63 Jun  8 16:19 text3
-rw-rw-r--  1 ian ian   26 Jun  8 16:19 text4
-rw-rw-r--  1 ian ian   24 Jun  8 16:42 text5
-rw-rw-r--  1 ian ian   98 Jun  8 17:09 text6
-rw-rw-r--  1 ian ian   15 Jun  8 15:48 xaa
-rw-rw-r--  1 ian ian    9 Jun  8 15:48 xab
```

Code	Object type
–	Regular file
d	Directory
l	Symbolic link
c	Character special device
b	Block special device
p	FIFO
s	Socket

r, w и x за директории
11-ти символ space, "." или "+"

Примери

```
$ chmod a= ab.txt
---------- 1 stoyan stoyan 0 фев 25 23:40 ab.txt
$ chmod u=xrw,g=rw,o+r ab.txt
-rwxrw-r-- 1 stoyan stoyan 0 фев 25 23:40 ab.txt
```

- **chmod**
  - *chmod +rwx file_name*
  - *chmod ug+rw file_name*
  - *chmod o-xrw file_name*
  - *chmod ug=rwx,o=␣ file_name*

- *Тест с файл*
  - *echo 'echo "Hello everybody!"'>hello.sh*
  - *ls -l hello.sh*
  - *./hello.sh*
  - *chmod +x hello.sh*

Symbolic	Octal	Binary
rwx	7	111
rw-	6	110
r-x	5	101
r--	4	100
-wx	3	011
-w-	2	010
--x	1	001
---	0	000

Задаване на собственик и група на файла

- Смяна на собственика chown – изпълнява се с root права.
  - **chown** <user name>[:][<group name>] <name>
  - **chown** :<group name> <name>
    - ако са зададени потребителско име и група, се сменят и двете;
    - ако е зададено само потребителско име, се сменя само собственика;
    - ако е зададено потребителско име и ":" файлът се прехвърля към основната група на потребителя;
    - ако е зададено ":" и група, собственикът не се променя;
- Смяна на групата **chgrp**
  - **chgrp** <group name> <name>
  - може да се изпълни като нормален потребител, ако потребителят е собственик на файла и член на групата към която го прехвърля.
- Смяната на собственика или на групата на файла не променя правата за достъп в различните категории (собственик, група, други).
- И двете команди поддържат опцията "–R"
- **newgrp** – прехвърля потребителя към друга група, към която принадлежи. Създава се нов шел и потребителят работи в него. Командата exit ни връща към предната група.

Задаване на права по подразбиране **umask**

- Стандартно при създаването на файл се изискват 0666, а за директории 0777 права.
- Използва се специална стойност (**umask**), с която да се настройва, какви права да **не** се дават за новосъздадени файлове и директории
- **umask** и **umask –S** – показват текущия статус, съответно в цифров или буквен вид
- **umask u=rwx,g=,o=** задава нови права;
- Когато се задава с цифри, се задават тези права, които не трябва да има файла;
- Може да се настрои в профилния файл (~/.profile, ~/.bash_profile или ~/.bashrc)
- Влияе на **chmod**, когато не се указва, за кого се прилага правилото:
  - **chmod** +x file

1.	Umask value:	027	----w-rwx
2.	Complement of umask value:	750	rwxr-x---
3.	A new file's access mode:	666	rw-rw-rw-
4.	Result (2 and 3 ANDed together):	640	rw-r-----

$$666 \ominus 027 = 640$$

$$6 \ominus 0 = 6,\ 6 \ominus 2 = 4,\ 6 \ominus 7 = 0$$

Специални режими на достъп

- **uid** и **guid** при **login**
  - Всички стартирани програми ги наследяват и с тях, и съответните права за достъп
- Специални режими
  - **suid** (set user id) (4) (работи само на binary програми, не за скриптове)
  - **sgid** (set group id) (2)
- Командата **passwd** и файл **/etc/password**

```
ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 30768 Feb 22  2012 /usr/bin/passwd
```

Програма с такъв флаг действа все едно, че я изпълнява собственика на файла

- chmod u+s g-s file_name

s (при изпълними файлове) и S (при останалите файлове)

chmod 4677 (s или S)?

Директории и sgid

- При зададен **sgid** режим за директория, всички файлове или директории създадени в нея наследяват нейното group id, т.е. все едно, че всички файлове са от групата на директорията;
- Правата за достъп до директорията не се променят, т.е. за да създадете файл, пак са необходими същите разрешения;
- Файл създаден в такава директория принадлежи към групата на директорията, независимо че създателят му може изобщо да не е в тази група;
- Този начин на работа обикновено се използва, за работни директории за проект, в който участват няколко човека.

Sticky bit

- Последният свободен бит за управление на достъпа
- Когато е зададен за директория позволява само на собственика на директорията или на root да изтрива файлове в нея
  - **chmod +t** directory
  - **ls –ld** directory
- *Съвременните версии на Linux игнорират този бит, ако е зададен за файл*

Access mode	Symbolic	Octal
suid	s with u	4000
sgid	s with g	2000
sticky	t	1000

Извеждане на осмичното представяне на режима на достъп

- Не се извежда от **ls -l**
- find . -name **file_name** -printf "%M%m%f\n"

Атрибути на файловете (изискват се root права)

Attribute	Meaning
A	*atime* is not updated (interesting for mobile computers)
a	(*append-only*) The file can only be appended to
c	The file's content is compressed transparently (not implemented)
d	The file will not be backed up by `dump`
i	(*immutable*) The file cannot be changed at all
j	Write operations to the file's content are passed through the journal (`ext3` only)
s	File data will be overwritten with zeroes on deletion (not implemented)
S	Write operations to the file are performed "synchronously", i. e., without buffering them internally
u	The file may be "undeleted" after deletion (not implemented)

- **chattr +i file_name** такъв файл не може да бъде изтрит дори от root
- **lsattr file_name** – за да проверим стойността

- Благодаря за вниманието!
- Въпроси?